Internet Draft S. Berson
Expires December 24, 1999 ISI
File: draft-berson-rsvp-ipv6-fl-00.txt July 1999

RSVP and Integrated Services with IPv6 Flow Labels

June 24, 1999

Status of Memo

Abstract

Currently, there is a brief definition about how RSVP works with IPv6[2]. However, there are a number of recent issues that need to be elaborated. These include issues with changes in the IPv6 Flow Label definition, routing RSVP messages, and security.

1 Introduction

There are potential significant benefits to the use of IPv6 flow labels with RSVP and integrated services. For layering reasons, it would be preferred if reservations were based only on the network layer. Since the flow label field is in the IPv6 header, RSVP reservations for IPv6 can be based only on fields in the IP header.

This draft addresses several issues that are related to the use of RSVP with IPv6 and flow labels. First, a new FILTER_SPEC format needs to be specified that contains both the IPv6 flow label and the source port. In addition, there has been a change in the format of a flow label in IPv6; RSVP needs to reflect this new format in its IPv6 Flow Label FILTER_SPEC. Second, there is no IPv6 API specified for an RSVP implementation to acquire a flow label. Third, the routing of a packet may depend on the flow label (as well as other fields). If RSVP is to route a message the same way as data packets are routed, the RSVP interface to routing will need access to the flow label. Finally, there are security issues where a node can spoof a flow label (and source address) to get preferred service or for a denial-of-service attack.

For information about IPv6 and flow labels including updated field definitions, see [3]. For information about a sockets interface to IPv6 including the use of flow labels, see [5, 9]. For general information about how flow labels are meant to be used, see [7]. Some of the ideas in this draft were originally described in [8].

2 Change of IPv6 Flow Label FILTER_SPEC Format

We expect that IPv6 with flow labels will be deployed incrementally. This means that an RSVP flow may encounter both routers that support IPv6 flow labels and routers that do not. If an RSVP reservation is to provide end-to-end reservations in this environment, it is necessary to provide a source port as well as a flow label in a filterspec. Thus, a FILTER_SPEC (and a SENDER_TEMPLATE) object needs to be defined that contains both an IPv6 flow label and a source port.

Also, since the RSVP specification first came out, the flow label field has been shortened from 24 bits to 20 bits[3]. This needs to be reflected in the RSVP specification.

To support this, the following needs to be appended to section A.9 of the RSVP specification:

2.1 New Filterspec Object

```
o    IPv6 Flow Label FILTER_SPEC object: Class = 10, C-Type = 6

     +-------------+-------------+-------------+-------------+
     |                                                       |
     +                                                       +
     |                                                       |
     +                IPv6 SrcAddress (16 bytes)             +
     |                                                       |
     +                                                       +
     |                                                       |
     +-------------+-------------+-------------+-------------+
     |   //////    |    //////   |         SrcPort           |
     +-------------+------+------+-------------+-------------+
```

```
|   ///////////////  |  Flow Label (20 bits)           |
+-------------+------+------+------------+-------------+

      Flow Label

           A 20-bit Flow Label, defined in IPv6.  This value may be used
           by the packet classifier to efficiently identify the packets
           belonging to a particular (sender->destination) flow.
```

There is a corresponding new format for a SENDER_TEMPLATE object. To support this the following text should be added to section A.10 of the RSVP specification:

```
o IPv6 Flow label SENDER\_TEMPLATE object: Class = 11, C-Type = 6

Definition same as IPv6 flow label FILTER\_SPEC object.
```

These new FILTER_SPEC and SENDER_TEMPLATE objects logically replace the old FILTER_SPEC and SENDER_TEMPLATE objects, i.e. Class=10, C-Type=3 and Class=11, C-Type=3, and so the relevant text from sections A.9 and A.10 should be deleted.

Note that there is a missing sentence in section A.10 that should have been in the RSVP specification that is now obsolete.

```
o IPv6 Flow label SENDER\_TEMPLATE object: Class = 11, C-Type = 3

**** Definition same as IPv6 flow label FILTER\_SPEC object.
```

3 Flow Label Allocation

In the definition of an IPv6 Flow Label[3], there are two main requirements on how a flow label is chosen. First, a flow label needs to be unique per sender, and second, the flow label needs to be pseudo-random (and uniform). The current sockets interface definition[5, 9] does not define a mechanism for establishing flow labels (draft versions of these RFCs specified mechanisms for use of flow labels, but this has been removed for the RFC). Clearly, the flow label should be chosen by the operating system (and not by the application) to ensure both the uniqueness and the randomness, but the details of the interface need to be completed. The main requirement for RSVP is that this flow label be accessible to RSVP, since RSVP needs to include the flow label in PATH messages (as well as to properly route the PATH message as described in section 4. There is an analogous problem with the GPI used for IPSEC data flows[1].

There are several different approaches for providing an IPv6 flow with an IPv6 flow label. The simplest approach is to provide a system call that allocates a new flow label. This flow label can then be used in the flowinfo field of the sockaddr6 structure for the destination as well as passed to RSVP. The problem with this approach is that assigning labels to packets should be a system function, not a user function.

An alternative approach is for the application to request that a socket use flow labels, e.g. by using a setsockopt(). For flow labels, a bind call should be required to use flow labels, and then a flow label would be associated with the bound socket. It would still be necessary to provide a call (e.g. a getsockopt) for the application to extract the flow label from a socket so that the flow label can be provided to RSVP.

A final approach is for the kernel to automatically start using a flow label when flow-like behavior is observed. In this case, the kernel would need to have an RSVP upcall that would pass the current RSVP flow descriptor (with source and destination addresses, source and destination ports, and protocol) to the RSVP daemon. RSVP would then need to locate the correct reservation and add the flow label field.

4 IPv6 Flow Labels and Routing RSVP messages

RSVP is intended to route RSVP signalling messages along the same path (forward or reverse) used by data messages for the same session. This is accomplished by using a routing interface that has been defined for RSVP[10] that allows RSVP access to kernel routing information. Currently this interface includes primarily source and destination address. The definition of the IPv6 flow label allows routing to be based on the flow label as well as source and destination addresses. In order to support proper routing of RSVP messages for IPv6 flow label sessions, the flow label must be included in the RSRR interface.

Routing of messages may be affected by other fields in the IP header that are currently not part of the RSRR interface. An IPv4 packet can have a source route listed as an IP option. Routing might also be affected by the Type-Of-Service bits in the IP header now being used for differentiated services[6]. In these and potentially other cases, the additional information should be provided to RSRR from RSVP in order to correctly route RSVP messages. For example, this may require passing a complete IP header to RSRR.

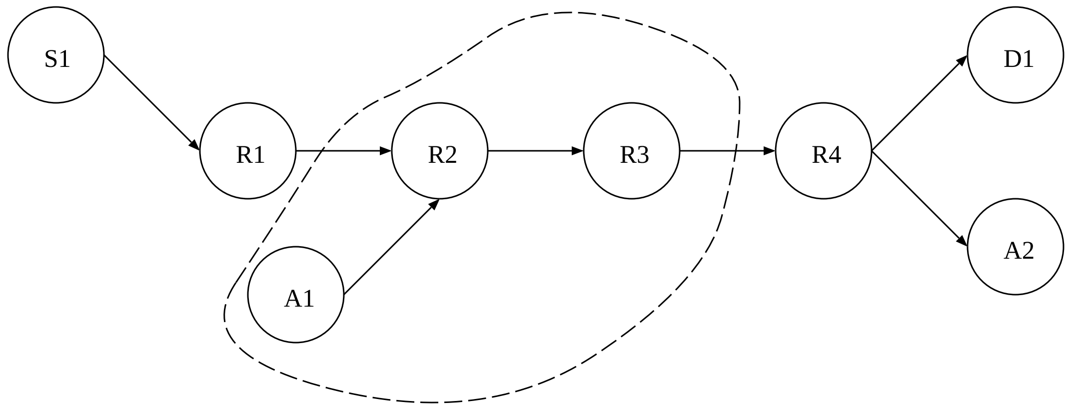

Figure 1: Security attacks

5 Security Issues

According to the IPv6 specification[3], a router does not need to look at the IPv6 destination address to route a packet. Thus a packet at an IPv6 router might be forwarded based only on the IPv6 source address and the flow label. Similarly an RSVP reservation can provide preferred service for a certain IPv6 source address and flow label pair. Thus using RSVP with IPv6 flow labels, spoofing of the source address and flow label potentially allows both theft-of-service and denial-of-service attacks.

For a denial of service attack, an attacking router would inject packets carrying a spoofed source address and flow label. These packets would need to arrive at one of the routers along the reserved data path. Once the packets reach the data path, the packets will follow the path as long as forwarding is done by source address and flow label. To understand this better, consider figure 1. Figure 1 shows several routers including $R2$ and $R3$ that use flow labels and source address (but not destination address) for routing. Destination $D1$ has a IPv6 flow label reservation for traffic from source $S1$ using flow label f. To deny service to $D1$, an attacker $A1$ sends data spoofing the address of $S1$ and the flow label f.

A similar approach can be used for theft of service. In this case, the attacker $A1$ wants to send reserved data to $A2$. $A1$ spoofs $S1$'s source address and the flow label f and sends data with destination address $A2$. The data will follow the flow label path going from $R2$ to $R3$ to $R4$. Since $R4$ is outside the IPv6 flow label routing cloud, the packets would then be forwarded to $A2$. This allows $A2$ to receive reserved service over the $R2$-$R3$-$R4$ links without having a reservation, essentially stealing reserved service over these links.

One approach to solving these security problems is to keep the identity of flow labels secret. With random assignment in a a 20 bit space, guessing a flow label will be difficult. However, the flow label must be exposed to all ISPs along the data path. An alternate approach for this security problem is for packet scheduling to filter on destination address as well as source address and flow label, but this removes the potential advantage of using flow labels. A final approach would be to periodically change flow labels for the same flow similar to the way keys are changed in [1]. This approach works well with secret keys, but likely will not work well with flow labels as flow labels are sent in the clear in IP messages. Thus an attacker will probably be able to acquire the new flow label as it changes.

6 Conclusions

In order for IPv6 with flow labels to be useful with RSVP, several steps must be taken. The first is to complete the sockets interface for use with IPv6 flow labels including support for RSVP to acquire the flow label. The second is to adapt RSRR to handle flow labels. The third is to make two technical changes to the RSVP specification to deal with 20 bit IPv6 flow labels and filters with both flow label and source ports. The final is to solve the security problems related to spoofing source addresses and flow labels.

References

[1] Berger, L., O'Malley, T., "RSVP Extensions for IPSEC Data Flows," RFC 2207.

[2] Braden, R., Zhang, L., Berson, S., Herzog, S., and Jamin, S., "Resource ReSerVation Protocol (RSVP) – Version 1 Functional Specification," RFC 2205.

[3] Deering, S., Hinden, R., "Internet Protocol, Version 6 (IPv6) Specification," RFC 2460.

[4] Gan, D., Guerin, R., Kamat, S., Li, T., Rosen, E., "Setting up Reservations on Explicit Paths using RSVP," Internet Draft.

[5] Gilligan, R., Thomson, S., Bound, J., Stevens, W., "Basic Socket Interface Extensions for IPv6," RFC 2553.

[6] Nichols, K., Blake, S., Baker, F., Black, D., "Definition of the Differentiated Services Field (DS Field) in the IPv4 and IPv6 Headers," RFC 2474.

[7] Partridge, C., "Using the Flow Label Field in IPv6," RFC 1809.

[8] Schmid, S., Dunmore, M., Scott, A., "RSVP Extensions for IPv6 Flow Label Support," Internet Draft.

[9] Stevens, W., Thomas, M., "Advanced Sockets API for IPv6," RFC 2292.

[10] Zappala, D., Kann, J., "RSRR: A Routing Interface For RSVP," Internet Draft.

Security Considerations

See section 5.

Author's Address

Steven Berson
USC Information Sciences Institute
4676 Admiralty Way
Marina del Rey, CA 90292

Phone: +1 310 822 1511
EMail: berson@isi.edu